Difference Between Call by Value and Call by Reference

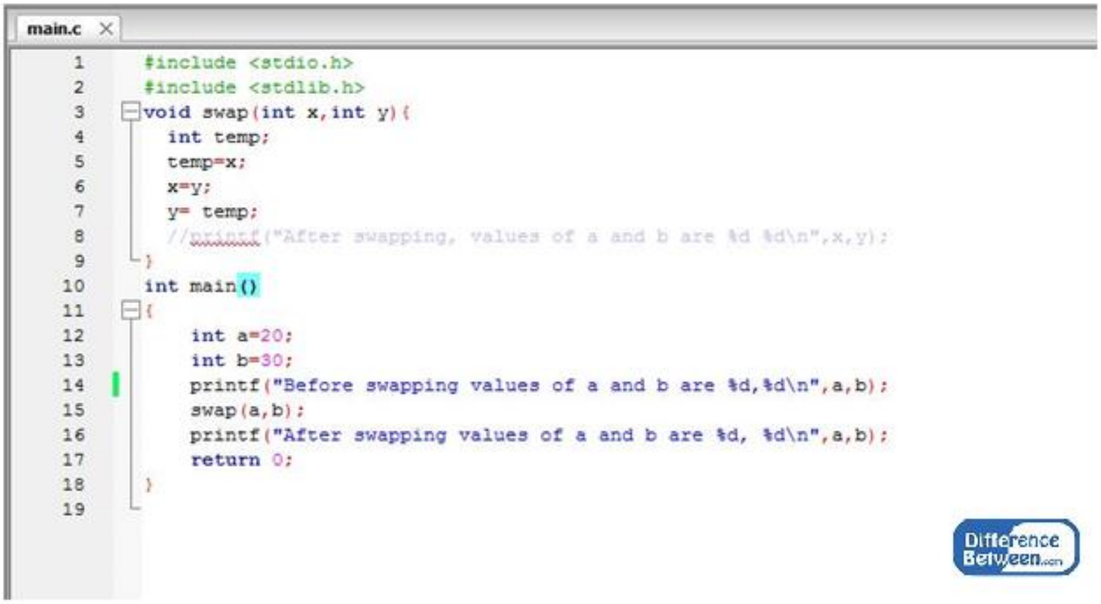
www.differencebetween.com

Key Difference - Call by Value vs Call by Reference

Many programming languages use functions. A function is a set of statements to perform a specific task. The main code can be divided into several functions and call them. There are two ways of calling a function such as call by value and call by reference. In call by value method, copies of variables are passed into the function. If there is a change in the value of the function, it does not change the original value. In call by reference, changes of the variable inside the function, reflect the original value. The **key difference** between call by value and call by reference is that**, in call by value, the copy of variables are passed to the function and in call by reference, the addresses of the variables are passed to the function**. This article discusses the difference between call by value and call by reference.

What is Call by Value?

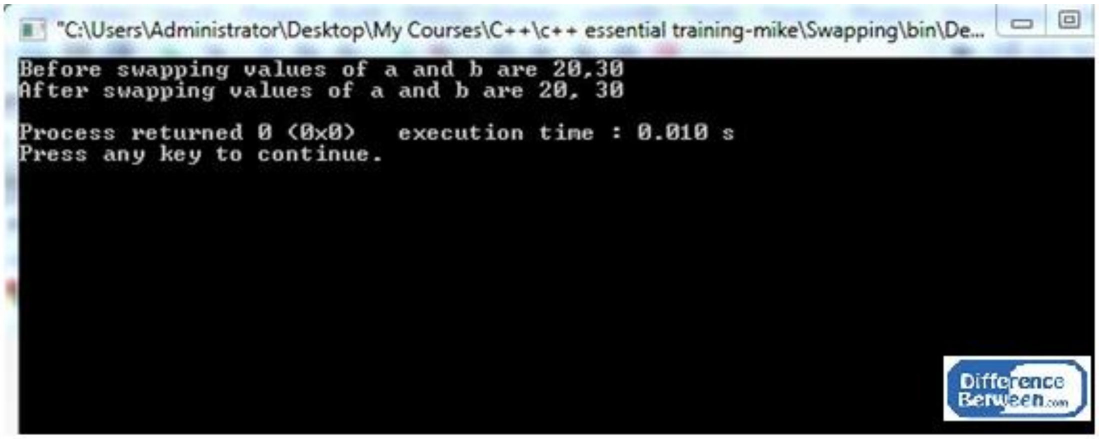
In call by value, copies of variables are passed to the function. The function may change that copied value but it does not change the original value. This concept can easily be understood by swapping two numbers program. Refer the below C program written using Code Blocks environment.



```
#include <stdio.h>
#include <stdlib.h>
void swap(int x,int y){
  int temp;
  temp=x;
  x=y;
  y= temp;
  //printf("After swapping, values of a and b are %d %d\n",x,y);
}
int main()
{
    int a=20;
    int b=30;
    printf("Before swapping values of a and b are %d,%d\n",a,b);
    swap(a,b);
    printf("After swapping values of a and b are %d, %d\n",a,b);
    return 0;
}
```

Figure 01: Program to Swapping two Numbers

According to the program given below, the variable 'a' has value 20 and variable 'b' has value 30. When calling swap(a,b) function, these two values are passed to the swap function. In the swap function, 'a' is copied into 'x' and 'b' is copied into 'y'. Now, 'x' is having value 20 and 'y' is having value 30. Using the temp variable the two values are swapped. When printing the value of 'x' and 'y' inside that swap function, the output will give 30 for 'x' and 20 for 'y'. When the swap function execution is over, the control returns back to the main function. When printing the values of 'a' and 'b', in the main function, the results will be the original values. They are 20 for 'a' and 30 for 'b'. Changes made using swap function does not reflect in the main program.


"C:\Users\Administrator\Desktop\My Courses\C++\c++ essential training-mike\Swapping\bin\De...

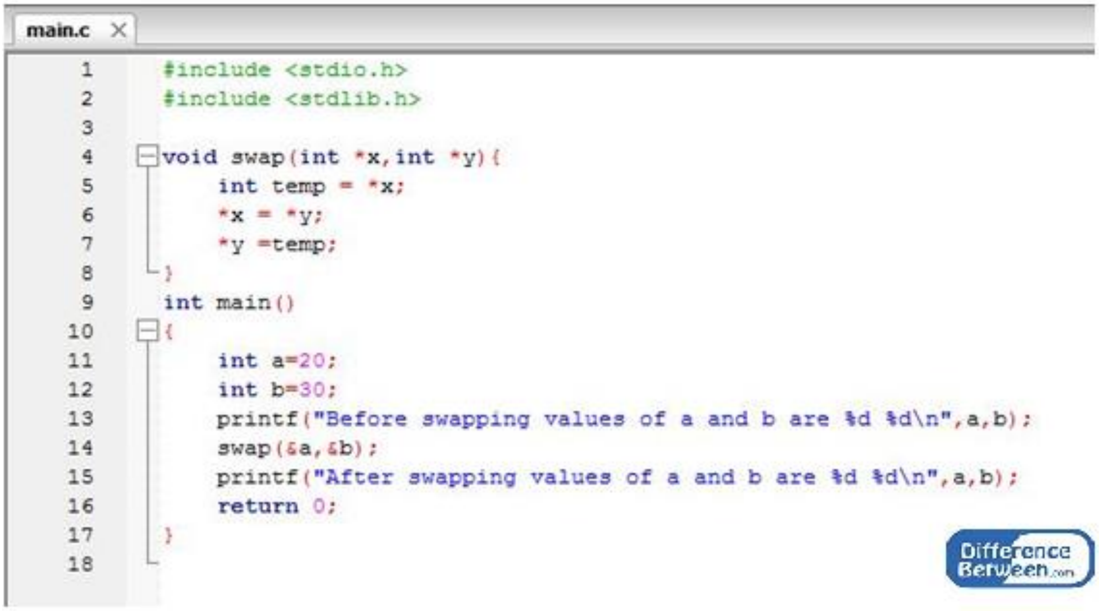
```
Before swapping values of a and b are 20,30
After swapping values of a and b are 20, 30

Process returned 0 (0x0)   execution time : 0.010 s
Press any key to continue.
```

Figure 02: Output of the Swapping Program

'x' and 'y' inside the swap function has the swapped values which are 30 and 20 but those values cannot be used in the main program. Even though 'x' and 'y' are having swapped values, they are local variables to swap function and cannot be accessed by the main function. To avoid this problem, call by reference can be used.

What is Call By Reference?

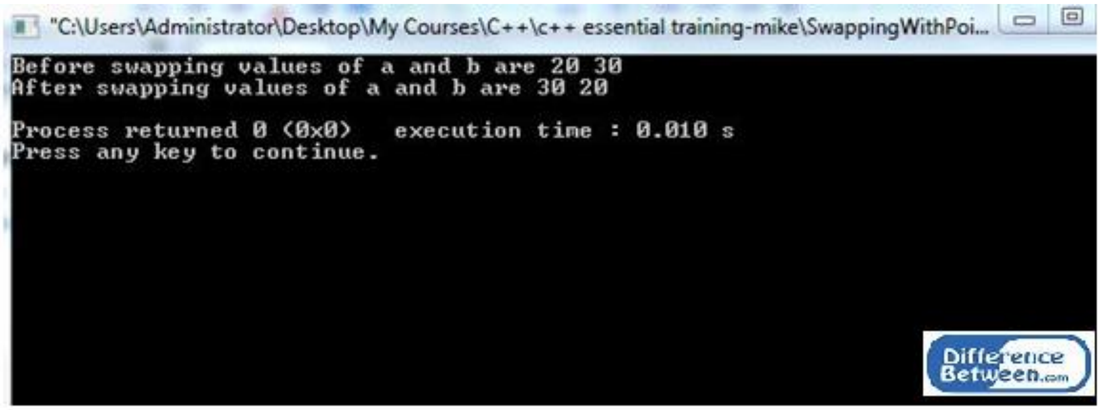
In this method, addresses of the variables are sent to the function. Refer the below C program written using Code Blocks environment.



```c
#include <stdio.h>
#include <stdlib.h>

void swap(int *x,int *y){
    int temp = *x;
    *x = *y;
    *y =temp;
}
int main()
{
    int a=20;
    int b=30;
    printf("Before swapping values of a and b are %d %d\n",a,b);
    swap(&a,&b);
    printf("After swapping values of a and b are %d %d\n",a,b);
    return 0;
}

```

Figure 03: Swapping of two Numbers using Pointers

According to the given example below, the value of 'a' is 20 and value of 'b' is 30. Instead of passing copies of 'a' and 'b', the programmer can send the addresses of 'a' and 'b' using the function, swap(&a, &b).



"C:\Users\Administrator\Desktop\My Courses\C++\c++ essential training-mike\SwappingWithPoi...

```
Before swapping values of a and b are 20 30
After swapping values of a and b are 30 20

Process returned 0 (0x0)   execution time : 0.010 s
Press any key to continue.
```

Figure 04: Output of the Swapping Program using Pointers

In the swap, the function should use pointers because addresses are passed and those addresses should be held by pointers. Inside the function, the address of 'a' is copied to 'x' and address of 'b' is copied to 'y'. Using the temp variable, 'x' and 'y' values are swapped. In this call by the reference method, the changes inside the swap function reflect in the main program. Printing the values of 'a' and 'b' in the main will give the swapped values. Now the output of 'a' is 30 and 'b' is 20. As the function is taking the address of the variables, the changes made inside the function affects the original values.

What is the Similarity Between Call By Value and Call By Reference?

- These two methods are used for calling functions/methods.

What is the Difference Between Call By Value and Call By Reference?

Call By Value vs Call By Reference	
In call by value, copies of variables are passed into the function so changes made inside the function, will not modify the original value.	In call by reference, the address of the variables is passed to the function, so changes made for variables inside the function, will modify the original value.
Value Modification	
In call by value, the original value is not changing.	In call by reference, the original values are changing.

Summary - Call by Value vs Call by Reference

Call by value and call by reference are methods of calling the function. The difference between call by value and call by reference is that in the call by value the copies of variables are passed to the function and in the call by reference, the addresses of the variables are passed to the function. Using call by value or call by reference depends on the task to perform.

Reference:

1.Jaiswal, Haresh . “11 Call by Value vs Call by Reference.” *YouTube*, YouTube, 24 Dec. 2015. Available here

How to Cite this Article?

APA: Difference Between Call by Value and Call by Reference.(2017 December 26). Retrieved (date), from http://differencebetween.com/difference-between-call-by-value-and-vs-call-by-reference/

MLA: "Difference Between Call by Value and Call by Reference" Difference Between.Com. 26 December 2017. Web.

Chicago: “Difference Between Call by Value and Call by Reference”. Difference Between.Com. http://differencebetween.com/difference-between-call-by-value-and-vs-call-by-reference/accessed (accessed [date]).